AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A security module for use with a terminal, comprising:

a data interface adapted to be coupled to a terminal, ~~for receiving~~configured to receive ~~one of~~ a first part of an algorithm code ~~and the complete algorithm code~~ from the terminal, with the algorithm code concerning a processing of secrets,

a power interface ~~for receiving~~configured to receive power from the terminal;

a volatile memory ~~for storing the one of~~ configured to store the first part of the algorithm code ~~and the complete algorithm code~~ received via the data interface, said volatile memory being coupled to the power interface in order to have power supplied thereto such that said volatile memory will be cleared upon an interruption of the receipt of the power from the terminal; and

a non-volatile memory in which a second part of the algorithm code which is a non-received remainder of the algorithm code is stored, wherein the first and second parts of the algorithm code form the complete algorithm code; and

a processor ~~for performing~~configured to perform the algorithm code in order to obtain an algorithm code result that can be delivered to the terminal,

wherein the first part of the algorithm code consists of memory addresses of computing components necessary for performing the algorithm code, and/or jump addresses of the algorithm code pointing to partial routines within the second part of the algorithm code.

2. (Currently Amended) A security module according to claim 1, wherein the data interface is adapted to receive only the first part of the algorithm code~~, the security module further comprising:~~

~~a non-volatile memory in which a remainder of the algorithm code which, along with the received part of the algorithm code, forms the complete algorithm code, is stored~~.

3. (Original) A security module according to claim 1, further comprising:

a means for performing an authentication between the terminal and the security module.

4. (Currently Amended) A security module according to claim 1, wherein the data interface is arranged to receive from the terminal the ~~one of the~~first part of the algorithm code ~~and the complete algorithm code~~ in encrypted form and with a certificate, with the security module further comprising:

a means for decrypting the ~~one of the~~first part of the encrypted algorithm code and the ~~encrypted~~

complete algorithm code; and

a means for examining the certificate and for preventing performing of the algorithm code depending on the examination of said certificate.

5. (Canceled)

6. (Previously Presented) A security module according to claim 1, further comprising:

a means for monitoring a predetermined security condition and for clearing the volatile memory if said predetermined security condition is fulfilled, with said security condition being

selected from an interruption of a supply voltage, a fluctuation of the supply voltage and an interruption of a system clock.

7. (Previously Presented) A security module according to claim 1, wherein the algorithm code comprises a program code selected for carrying out a task selected from the group consisting of a symmetric cryptographic algorithm, an asymmetric cryptographic algorithm, an RSA algorithm, a cryptographic process according to the DES standard, an elliptic curve process, an access function for accessing a digital value stored on the security module and an access function for changing the digital value stored on the security module.

8. (Canceled)

9. (Currently Amended) A security module according to claim 1, wherein the data interface is adapted to receive the ~~one of the~~first part of the algorithm code ~~and the complete algorithm code~~ several times in different versions, with the volatile memory being arranged for being overwritten by the different versions of ~~one of the~~first part of the algorithm code ~~and the complete algorithm code~~ at the several times.

10. (Previously Presented) A security module according to claim 1, wherein said security module is designed as a chip card.

11. (Currently Amended) A process for computing an algorithm code result using a security module, comprising ~~the steps of~~:

receiving ~~one of~~ first part of an algorithm code ~~and the complete algorithm code~~ by means of an interface to a terminal, with the algorithm code concerning a cryptographically processing of data~~processing of secrets~~;

storing the ~~one of the~~first part of the algorithm code ~~and the complete algorithm code~~ in a volatile memory of the security module, with the volatile memory being coupled to the interface, to be supplied with power, such that the volatile memory will be cleared upon an interruption of the receipt of the power from the terminal, wherein a second part of the algorithm code is a non-received remainder of the algorithm code is stored in a non-volatile memory of the security module;

performing said algorithm code on the security module in order to obtain cryptographically processed data~~an algorithm code result~~; and

delivering said cryptographically processed data ~~algorithm code result~~ to the terminal,

wherein the first part of the algorithm code consists of memory addresses of computing components necessary for performing the algorithm code, and/or jump addresses of the algorithm code pointing to partial routines within the remainder of the algorithm code.

12. (Previously Presented) A process according to claim 11, further comprising removing the security module from the terminal thereby causing an interruption of the receipt of the power to the volatile memory from the terminal and clearing said volatile memory upon interruption of the receipt of power from the terminal.

13. (Currently Amended) A terminal for use with a security module having a volatile memory being able to be supplied by power from the terminal, such that the volatile memory will be cleared upon an interruption of a supply of the power, and having a non-volatile memory in which a second part of an algorithm code is stored, comprising:

a data interface adapted to be coupled to the security module, for transmitting ~~at least~~ a first part of ~~an~~ the algorithm code ~~or the complete algorithm code~~ from the terminal to ~~a~~ the volatile memory of the security module and for receiving ~~an algorithm code result~~ cryptographically

processed data from the security module, with the algorithm code concerning a processing of secrets, wherein the second part is a remainder of the algorithm code; and

a power interface ~~for delivering~~adapted to deliver power to the security module, with the volatile memory being supplied by the power, such that the volatile memory will be cleared upon an interruption of the receipt of the power from the terminal,

for each communication operation between the terminal and the security module, the data interface controlled to: send ~~at least~~ the first part of the algorithm code ~~or the complete algorithm code~~ to the volatile memory of the security module and then to receive the ~~algorithm code result~~ cryptographically processed data from the security module,

wherein the first part of the algorithm code consists of memory addresses of computing components necessary for performing the algorithm code, and/or jump addresses of the algorithm code pointing to partial routines within the remainder of the algorithm code.

14. (Currently Amended) A process for controlling a security module using a terminal in order to obtain cryptographically processed data ~~an algorithm code result~~ from the security module, the process comprising ~~performing the following steps during~~for each one of a plurality of communication operations between the terminal and the security module:

delivering power from the terminal to the security module;

transmitting ~~at least~~ a first part of an algorithm code ~~or the complete algorithm code~~ from the terminal to a volatile memory of the security module, with the algorithm code concerning a cryptographic processing of ~~secrets~~data, with the volatile memory being supplied by the power, such that the volatile memory will be cleared upon an interruption of the receipt of the power from the terminal, and with the security module having a non-volatile memory in which a second part of the algorithm code which is a non-transmitted remainder of the algorithm code is stored; and

receiving the cryptographically processed data~~an algorithm code result~~ from the security module,

wherein the first part of the algorithm code consists of memory addresses of computing components necessary for performing the algorithm code, and/or jump addresses of the algorithm code pointing to partial routines within the remainder of the algorithm code.

15. (Currently Amended) A process for communication between a security module and a terminal, comprising~~ the steps of~~:

transferring ~~one of~~ a first part of an algorithm code ~~and the complete algorithm code~~ from the terminal to the security module, with the algorithm code concerning a cryptographic processing of ~~secrets~~data;

storing the ~~one of the~~first part of the algorithm code ~~and said complete algorithm code~~ in a volatile memory of the security module, with the volatile memory being supplied by power from the terminal, such that the volatile memory will be cleared upon interruption of the receipt of the power from the terminal, and with the security module having a non-volatile memory in which a second part of the algorithm code which is a non-received remainder of the algorithm code is stored;

performing said algorithm code on the security module in order to obtain~~ an algorithm code result~~ cryptographically processed data;

delivering said cryptographically processed data ~~algorithm code result~~ to the terminal; and

clearing said volatile memory upon an interruption of the receipt of the power from the terminal,

wherein the first part of the algorithm code consists of memory addresses of computing components necessary for performing the algorithm code, and/or jump addresses of the algorithm code pointing to partial routines within the remainder of the algorithm code.

16. (Currently Amended) A process according to claim 15, further comprising:

sequentially transferring a plurality of different versions of the ~~one of the~~first part of the algorithm code ~~and said complete algorithm code~~; and

sequentially storing the different versions of the ~~one of the~~ first part of the algorithm code ~~and the complete algorithm code~~ such that a respective previous version of the plurality of different versions of the ~~one of the~~ first part of the algorithm code ~~and the complete algorithm code~~ is overwritten.

17. (Currently Amended) A security module for use with a terminal, comprising:

a data interface adapted to be coupled to a terminal, for receiving a part of an algorithm code from the terminal, with the algorithm code concerning a cryptographic processing of data;

a power interface ~~for receiving~~configured to receive power from the terminal;

a volatile memory ~~for storing~~configured to store the part of the algorithm code received via the data interface, said volatile memory being coupled to said power interface in order to have power supplied thereto such that the volatile memory will be cleared upon an interruption of the receipt of the power from the terminal;

a non-volatile memory in which a remainder of the algorithm code which, along with the received part of the algorithm code, forms a complete algorithm code, is stored; and

a processor ~~for performing~~configured to perform the algorithm code in order to obtain cryptographically processed data that can be delivered to the terminal, wherein the part of the algorithm code includes memory addresses of computing components necessary for performing the algorithm code, ~~or~~ and/or jump addresses of the algorithm code pointing to partial routines within the remainder of the algorithm code.